

FIRE REPORT

Bishal

OVERALL PROJECT STRUCTURE

```
Fire-Classification/
├── configs/                # Hydra configs
│   ├── augment.yaml
│   ├── test.yaml
│   └── train.yaml
├── data/
│   ├── raw_fire_sample/    # raw fire sample images
│   ├── inference/          # sample image for inference
│   ├── fire/               # [generated] fire images
│   ├── non-fire/           # [generated] non-fire images
│   └── augmented/          # [generated] train/val/test splits
├── debug/
├── fireEnv/                # [generated] python environment
├── inference_outputs/      # [generated] inference output subdirs per run
├── outputs/                # [generated] output subdirs per run
├── saved_weights/          # Saved model weight
├── src/
│   ├── data/
│   │   └── dataloader.py    # Dataloader class used for training
│   ├── models/
│   │   ├── classification_model_scratch.py # custom model
│   │   ├── classification_model_yolo.py    # YOLO model
│   │   └── classification_model.py         # other pretrained models
│   ├── utils/               # contains utils files
│   ├── augment.py/          # data augmentation script
│   ├── train.py             # training script
│   ├── test.py              # evaluation (metrics, plots, CSV)
│   └── inference.py         # grid-based fire localization with saved json
```

DATA PREPROCESSING

1. Collected images from various sources online on fire related images.


MIVIA Fire Detection dataset 

Firenet Dataset 

Fire Flame Dataset 

SKLFS Dataset 

Kaggle Fire Dataset 

Roboflow Fire Datasets 

DATA PREPROCESSING

1. Extracting Fire and non-Fire image patches.



1	0	0.6863333333333334	0.6327916666666666	0.36574999999999996	0.5920000000000001
2	1	0.45369444444444446	0.5833333333333334	0.10647222222222222	0.5
3	1	0.48380555555555554	0.16408333333333333	0.8379722222222222	0.328125
4	1	0.9363333333333332	0.6041666666666666	0.12730555555555556	0.5208333333333334



Fire image patches



Non-Fire image patches

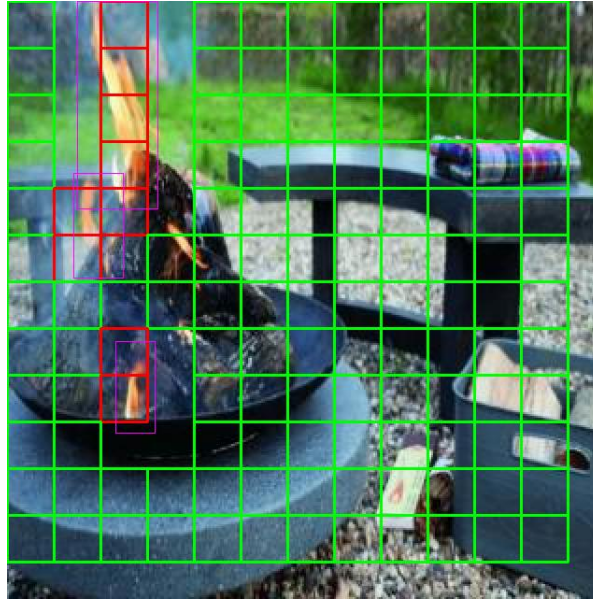
Make fire/non-fire image patches based on

DATA PREPROCESSING

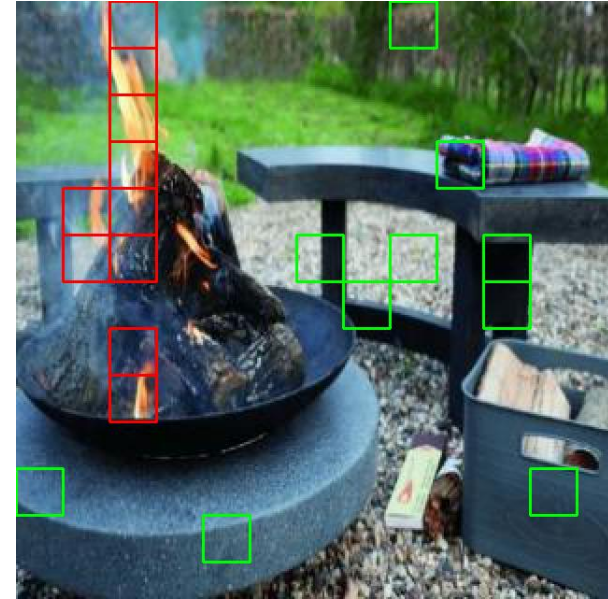
1. Extracting Fire and non-Fire image patches.



Original fire image with annotated ground-truth labels.



Divide the image into patches and classify each as fire (red) or non-fire (green) based on overlap with ground-truth fire regions.



Randomly select an equal number of non-fire patches to match the fire patches for balanced training data.

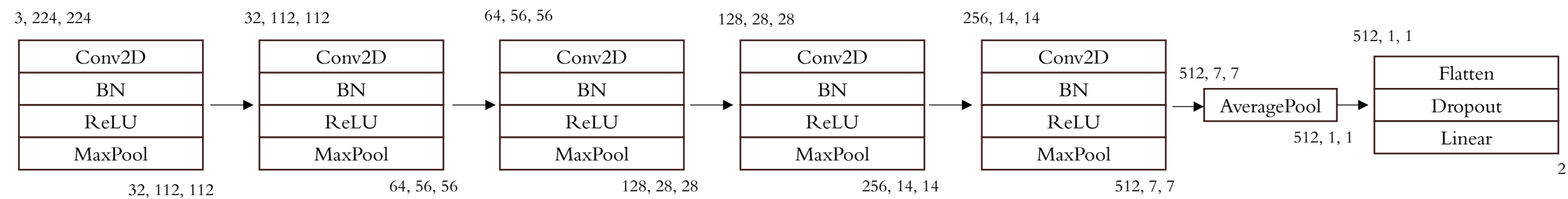
DATA PREPROCESSING

2. Divide into train, test and val. Perform augmentations on the train set.

```
# Albumentations pipelines (order = list order)
pipelines:
  train:
    - name: "HorizontalFlip"
      p: 0.5
    - name: "ShiftScaleRotate"
      shift_limit: 0.05
      scale_limit: 0.10
      rotate_limit: 15
      border_mode: 0
      value: [0, 0, 0]
      p: 0.5
    - name: "RandomBrightnessContrast"
      brightness_limit: 0.2
      contrast_limit: 0.2
      p: 0.5
    - name: "HueSaturationValue"
      hue_shift_limit: 10
      sat_shift_limit: 15
      val_shift_limit: 10
      p: 0.3
    - name: "GaussianBlur"
      blur_limit: 3
      p: 0.2
    - name: "GaussNoise"
      var_limit: [5.0, 20.0]
      p: 0.2
```

MODEL DETAILS

Fire Classification Model Structure



```
optim:
  name: "adam"
  lr: 0.0001
  weight_decay: 0.0
```

```
scheduler:
  name: "none"
```

```
training:
  epochs: 50
  batch_size: 32
  early_stop: true
  patience: 7
```

```
other:
  seed: 0
  log_print: true
  wandb: false
```

Other configurable parameters

FIRE CLASSIFICATION

Model Results in comparison to Resnet50 backbone model

Models	Accuracy	Precision	Recall	F1	Latency (ms/img)	Throughput (img/s)
Resnet50★	94.5	94.3	95.6	91.5	3.684	271.5
Fire Classifier	89.3	88.9	86.8	87.8	0.440	2270.8

★ Trained as model backbone



Resnet50 Output



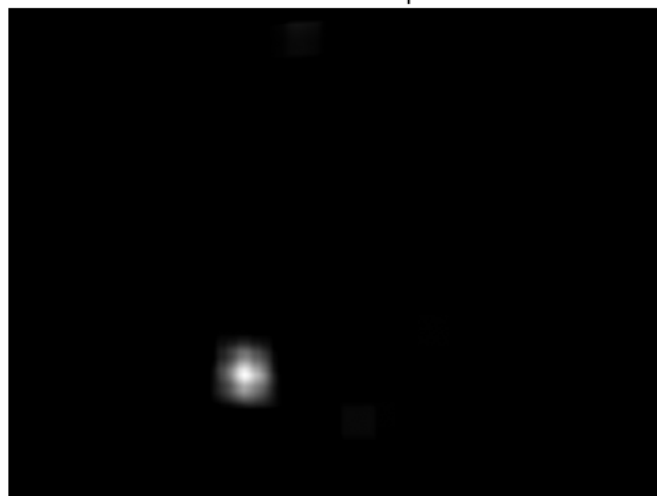
Fire Classifier Output

FIRE CLASSIFICATION

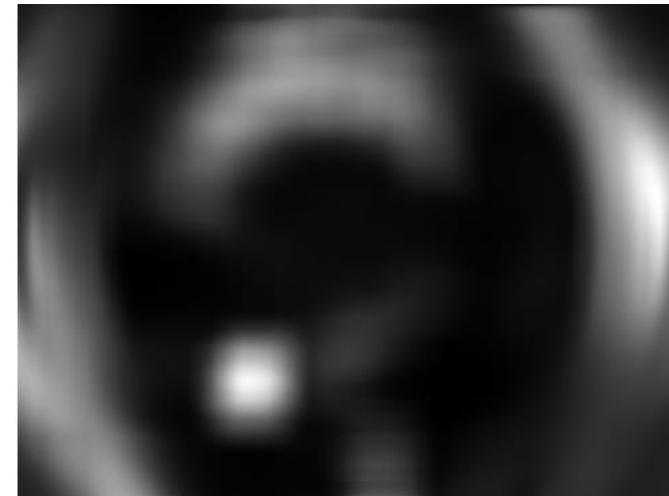
Model Results in comparison to Resnet50 backbone model



Input image



Resnet50 sliding window
classification heatmap



Fire classifier sliding window
classification heatmap

CODE SETUP AND USAGE

1. Create and activate environment

```
python -m venv fireEnv  
source fireEnv/bin/activate      # Linux / Mac  
fireEnv\Scripts\activate        # Windows
```

2. Install dependencies

```
pip install --upgrade pip  
pip install -e .
```

This installs the project in **editable mode**, so changes to the source code are immediately reflected without reinstalling.

CODE SETUP AND USAGE

4. Data Preprocessing

Generate augmented train/val/test splits (uses config file from `configs/augment.yaml`):

```
python augment.py
```

refer [preprocessing documentation](#) for more details.

5. Training

Train model (uses config file from `configs/train.yaml`):

```
python train.py
```

refer [training documentation](#) for more details.

6. Test

Run evaluation on the test set using config file from `configs/test.yaml`:

```
python test.py
```

Outputs:

- `metrics.csv` (accuracy, precision, recall, F1)
- Confusion matrix + classification report plots

refer [test evaluation documentation](#) for more details.

7. Inference (Grid-based detection)

Run fire localization on a new image:

```
python inference.py --model custom --model_path  
saved_weights/best_modelc.pth --img_dir data/inference/
```

Outputs:

- Saved to `inference_outputs/<timestamp>/`
- Includes original image, grid overlay, and results in json format.

refer [inference documentation](#) for more details.